Formal logic notation provides for the study of abstract mathematics what high school algebra provides for the solving of “story problems”. It allows for the expression of the ideas in a symbolic notation so that the expressions can then be manipulated to arrive at a solution to the problem. For this reason, we will take a brief look at this system, and will then use it as needed to clarify discussion of algebraic topics.

NOTATION

Propositions: A proposition is an expression in mathematical notation which expresses a complete idea, corresponding to a complete sentence in English. Examples that we will typically see are “\( x \in A \)”, “\( \alpha(x) = \alpha(y) \)”, or “\( (ab)^i = b^i a^i \)”. Expressions like “\( \sin(x^2 + y^2) \)” or “\( (A \cap B) \cup C \)” are not propositions because they do not express a complete idea. The simplest propositions, ones which correspond to simple sentences as opposed to compound sentences, are called atomic propositions or atomic formulas. We will represent propositions by letters \( P, Q, R, S, \ldots \) when we are discussing them in general.

Logical connectives for propositions: More complex propositions are built from simpler ones by combining them with logical connectives. Their notations and meanings are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>In Words / Meaning</th>
<th>Truth Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOT</td>
<td>( \neg P )</td>
<td>read “not ( P )”</td>
<td>\begin{tabular}{c</td>
</tr>
<tr>
<td>OR</td>
<td>( P \lor Q )</td>
<td>read “( P ) or ( Q )”</td>
<td>\begin{tabular}{c</td>
</tr>
<tr>
<td>AND</td>
<td>( P \land Q )</td>
<td>read “( P ) and ( Q )”</td>
<td>\begin{tabular}{c</td>
</tr>
<tr>
<td>IMPLICATION</td>
<td>( P \implies Q )</td>
<td>read “if ( P ) then ( Q )” or “( P ) implies ( Q )”</td>
<td>\begin{tabular}{c</td>
</tr>
<tr>
<td>BICONDIONAL</td>
<td>( P \iff Q )</td>
<td>read “( P ) if and only if ( Q )”</td>
<td>\begin{tabular}{c</td>
</tr>
</tbody>
</table>

When building complex propositions with these connectives, we often must use parentheses to eliminate ambiguity. For example, \( P \land Q \lor R \) could mean \( (P \land Q) \lor R \) or \( P \land (Q \lor R) \). The parentheses are needed to make it clear which grouping is intended since the meaning of the two possibilities are different.
Quantifiers: Most serious discussion of mathematics requires statements involving quantification. We often must state that something is true “for all” objects of a given sort or claim that “there exists” an object within a given universe of discourse that has a certain property. The notation set forth above is not adequate for expressing such ideas. To remedy this, we now add quantification symbols to the system as follows:

- $(\forall x)_S P$ will be read “for all $x$ in $S$, $P$ is true”
- $(\exists x)_S P$ will be read “there exists an $x$ in $S$ such that $P$ is true”

Generally in constructions such as this, $P$ will be a proposition which says something about $x$. When $P$ is a statement about $x$, we will use the notation $P(x)$ to emphasize this.

Examples:

1. The defining statement for a mapping $\alpha$ from a set $S$ to a set $T$ to be injective is “for every $x$ and $y$ in $S$, if $\alpha(x) = \alpha(y)$, then $x = y$.” This is rendered formally as

   $$(\forall x)_S (\forall y)_S (\alpha(x) = \alpha(y) \Rightarrow x = y)$$

2. The statement that every element in a group $G$ has an inverse element in $G$ can be represented formally as:

   $$(\forall x)_G (\exists y)_G (x \cdot y = e & y \cdot x = e)$$

3. A statement that for a mapping $\alpha$ from $S$ to $T$ every element of $T$ is the image of precisely two elements of $S$ could be rendered as:

   $$(\forall z)_T (\exists x)_S (\exists y)_S ((\neg x=y & \alpha(x) = z & \alpha(y) = z) & (\forall w)_S ((\neg w = x & \neg w = y) \Rightarrow \neg \alpha(w) = z))$$

Note what this says: For every $z$ in $T$ there are two unequal elements $x$ and $y$ in $S$ such that both map to $z$ under $\alpha$ and that any other $w$ in $S$ different from $x$ and $y$ does not map to $z$ under $\alpha$.

Negation of Statements

We said that formal notation is useful because it allows for the manipulation of complex ideas once they have been expressed formally. One of the most useful manipulation techniques involves the simplification of the negations of complex propositions. We say above in example (1) what it means for a mapping to be injective. What does it mean for a mapping to be not injective? When the statement in question is complicated enough, it is often helpful to process its negation in a mechanical way in order to better understand it. The basic idea is to reexpress the statement with the negation symbol pushed down to one or more atomic propositions, i.e., those that are not built from still simpler ones with logical connectives or quantifiers. This can be accomplished by repeated use of the following negation rules:
\[ \neg (\neg P) \text{ is equivalent to } P \]
\[ \neg (P \lor Q) \text{ is equivalent to } \neg P \land \neg Q \]
\[ \neg (P \land Q) \text{ is equivalent to } \neg P \lor \neg Q \]

or
\[ P \Rightarrow \neg Q \]

or
\[ Q \Rightarrow \neg P \]

\[ \neg (P \Rightarrow Q) \text{ is equivalent to } (P \land \neg Q) \lor (\neg P \land Q) \]

\[ \neg (P \iff Q) \text{ is equivalent to } (P \land \neg Q) \lor (\neg P \land Q) \]

\[ \neg (\forall x)_A P \text{ is equivalent to } (\exists x)_A \neg P \]

\[ \neg (\exists x)_A P \text{ is equivalent to } (\forall x)_A \neg P \]

**Examples**

(1) A mapping \( \alpha \) from \( S \) to \( T \) is not injective if

\[ (\exists x)_S (\exists y)_S (x \neq y \land \alpha(x) = \alpha(y)) \]

(2) If the statement “every element has an inverse” given formally above fails to hold, then its negation reduces to

\[ (\exists x)_G (\forall y)_G (x \cdot y \neq e \lor y \cdot x \neq e) \]

(3) “It is not the case that every element of \( T \) is the image under \( \alpha \) of precisely two elements of \( S \)” is the negation in English of the statement in example 3 above. This corresponds to simply putting a negation symbol in front of the formal version of the statement. If one uses one of the ways of pushing the “not” all the way down to an atomic proposition, one gets the formal statement

\[ (\exists z)_T (\forall x)_S (\forall y)_S ((x \neq y \land \alpha(x) = z \land \alpha(y) = z) \Rightarrow (\exists w)_S (w \neq x \land w \neq y \land \alpha(w) = z)) \]

Note that with the string of “&’s” to be negated on this last one, we have some choice as to how we do it. One in general gets a statement that will be easiest to work with by using the implication form and loading as much as possible onto the antecedent (left) side. The statement above can now be recast into English. It says that there is a element \( z \) in \( T \) with the property that for any \( x \) and \( y \) in \( S \), if \( x \) and \( y \) are unequal elements which both map to \( z \) under \( \alpha \), then there must be a third element \( w \) in \( S \) not equal to either of these which also maps to \( z \). This form of the negation of the original statement gives us new insight. It is “operational” in the sense that we can see how to prove it. Simply assume that \( \alpha \) maps two distinct elements of \( S \) to \( z \) and prove from this that a third element must exist which \( \alpha \) also maps to \( z \).

**NOTE:** We have used a common practice above for simplifying the negation of atomic propositions. One puts a slash through a relation symbol to indicate the negation of the relation. For example, the proposition “\( \neg x = y \)” is written as “\( x \neq y \)” and “\( \neg a < b \)” is written as “\( a \not< b \)”.